Fall 2011

CEG 361/561-01: Introduction to Software Testing

John A. Reisner
Wright State University - Main Campus, john.reisner@wright.edu

CEG 361/561: Introduction to Software Testing

Fall Quarter, 2011

Course Description

This course covers software testing strategies, along with established best practices, so students learn how to test their software in a complete and systematic (vice ad-hoc) manner. Particular attention is paid to planning, writing, and executing software tests, along with associated documentation, (i.e., a software test plan), which includes documented results. Various projects are assigned, designed to illustrate various challenges associated with software testing, and to reinforce the strategies and techniques used to overcome these challenges.

Textbook

Lee Copeland, *A Practitioner's Guide to Software Testing*, Artech House, 2004, ISBN 1-58053-791-X. This is a required textbook for this course.

Reading Assignments

Each week's lessons have corresponding reading assignments. The course lectures are designed to *augment* (not simply *rehash*) these readings.

The course text is a straightforward book. Chapters are written succinctly. It would behoove students to review the material in the book during the week when it is being covered in class.

Course Projects & Lectures

This will be a "learning-by-doing" class. Students will have a series of projects throughout the course, where they will write code, write test plans, execute test plans, and document the results. The course projects represent rather unique programming assignments, in that grading is based primarily on how well the code is ***tested***, as opposed to how well it is written.

Class time will consist of interactive discussions. Students should attend class ready to contribute through active participation. **No open laptops are allowed in class.** Class time is devoted to learning, not browsing.

Instructor Contact Info

John Reisner
Office hours after class or by appointment
Daytime Phone: 255-3636 x7422 (this is a WPAFB phone number).
email: **john.reisner@wright.edu** (for a more timely response, consider sending a CC: to **jreisner@afit.edu**).

The instructor is an adjunct faculty member. Most contact will be done via email, phone, or during before- or after-class discussions. Other meetings can be arranged as needed.

Course Objectives

Each student should be able to:

1. Write appropriately comprehensive test plans.
2. Effectively document test plans and results.
3. Develop software using a test-driven approach.
4. Employ effective testing strategies for different needs.
5. Write drivers, stubs, and testware as needed to sufficiently test a program.
6. Verify a program's correctness via a test strategy.

Course Components

Being a higher-level college course, my goal is to teach at the higher levels of **Bloom's Taxonomy**. Overall, the goal of this course's homework assignments, projects, and exams are to stimulate and promote ***thought***, particularly at higher levels of the cognitive and affective domains.

35% Course Projects

- These will be programming projects. The emphasis of these projects will be testing the software that has been tested, with a written test plan.
- All testing is to be performed using a written test plan, developed by the student.
- I may refer to these as "weekly" projects; however, in some cases, a project may be extended over two weeks, where students are expected to write the code during the first week, and execute the test plan during the second week. This is likely to happen when the test plans are expected to be exceptionally complex.
- Each project will be graded individually. Although the grade will be based primarily on the thoroughness and quality of the test plan, students are expected to use good programming style and employ accepted programming practices throughout the course.
- The student may select the programming language used for each project.
- Projects are to be submitted in paper form, to include code listings.

20% Midterm Exam

- Mixed-format exam, administered in class. Exams generally consist of 7 multiple-choice questions (28 points), plus some (usually between four and eight) short answer, essay, or analysis questions (72 points).
- Any of the material covered in assigned readings, or in-class discussions is considered "testable." That said, tests are designed to focus on and reiterate important and fundamental concepts, rather than minutiae.
- Many exam questions are designed to test the ability of the student to analyze, synthesize, and eloquently explain information and concepts, rather than recall simple facts. You may be asked to argue a certain position; your argument is expected to be sound, and demonstrate that you are learning in the course.

25% Final Exam

- Comprehensive, mixed-format exam, administered during the school's final exam week. This exam will resemble the midterm (although the questions will be different).

15% Homework Assignments

- Homework assignments are designed to facilitate deeper comprehension about a lecture topic (in other words, these are "think and respond" assignments).
- There may be up to two assignments per week, but some weeks may have one or zero assignments. Most weeks will not have more than one.
- Homework assignments are different from the weekly projects.
- Details about these assignments will be found on Pilot.
- Normally, these assignments will be due on Monday each week (thus, students have one week to complete a Monday assignment and five days to complete a Wednesday assignment). Any exceptions to this policy will be mentioned when the homework is assigned.

5% Class Participation

- Based on attendance, participation, attitude, readiness to learn, and willingness to share thoughts and ideas.

Grading of Course Work

All homework assignments and projects must be submitted in **paper form.**

At the end of the term, the following scale is used:

92-100 = A

84-91 = B

75-83 = C

65-74 = D

< 65 = F

However, this scale may be (and frequently is) curved, at the instructor's discretion, after all work has been graded, and the grade distributions have been analyzed.

Many of the assignments in this class will be graded subjectively, due to the nature of the work. Many assignments require turn-ins that are not necessarily *right* or *wrong*, but rather well- or poorly-documented, strongly or weakly substantiated, thorough or cursory, pithy or superficial, well-organized or carelessly compiled. Superior work is graded above 90; satisfactory work is graded between 80 and 90, and unsatisfactory work is graded below 80, depending upon the severity of the problems. Grading rubrics will be employed from time to time, to help students gain a better understanding of expectations.

Late Work

- No late work will be accepted after the last day of class for the quarter.
- For other late work, points are deducted based on how late the assignment is turned in, although extenuating circumstances will be considered. Work turned in just a few days after the deadline will have smaller deductions; deductions will increase (potentially substantially) thereafter. Advanced notice (via email) is considered a good practice.
- If you complete a late assignment several days before we meet for a class (e.g., say you complete an assignment on a Thursday; but we don't meet again until Monday), it is generally best to email me the work right away (this email will stop the "lateness penalty clock"). However, you must *also* print a copy, and hand in that hardcopy during the next class session (which prevents me from assessing the "instructor had to print this" penalty).

Course Schedule (subject to change)

Week	Lesson	Date	Lesson Topics	Reading Assignment	Project
1	1	Wed Sep _8_	Course Introduction Terminology & Basics Intro to Course Text Philosophies & Challenges	Chapters 1 & 2	Project 1 assigned. See details on Web CT.
2	2	Mon Sep _13_	Test Cases & Test Plans	Chapter 12	Project 1 due on Mon; Project 2 assigned. See details on Web CT.
	3	Wed Sep _15_	Testable Requirements The V-Model & Testing	Chapter 14	
3	4	Mon Sep _20_	Black-Box Testing Boundary Value Testing Equivalence Class Testing	Section I Introduction Chapters 3 & 4	Project 2 due on Mon; Project 3 assigned. See details on Web CT.
	5	Wed Sep _22_	Orthogonal Arrays	Chapter 6	
4	6	Mon Sep _27_	Testing for Robustness: Load, Performance, Stress, and Pathological Testing	Outside Readings	Project 3 due on Mon; Project 4 assigned. See details on Web CT.
	7	Wed Sep _29_	Decision Tables State Chart Testing	Chapters 5 & 7	
5	8	Mon Oct _4_	Test-Driven Development Use Case Testing	Chapter 9	
	9	Wed Oct _6_	Testing & the Software Lifecycle Integration Testing Strategies Large System Testing	Outside Readings	
6		Mon Oct _11_	MIDTERM EXAM		Project 4 due on Wed.
	10	Wed Oct _13_	Domain Analysis Testing	Chapter 8	
7	11	Mon Oct _18_	Exploratory Testing	Section III Introduction Chapter 13	Project 5 assigned. See details on Web CT.
	12	Wed Oct _20_	Intro to White-Box Testing	Section II Introduction	
8	13	Mon Oct _25_	Control Flow Testing	Chapter 10	
	14	Wed Oct _27_	Data Flow Testing (Static)	Chapter 11	
9	15	Mon Nov _1_	Data Flow Testing (Dynamic)		Project 5 due on Mon; Project 6 assigned. See details on Web CT.
	16	Wed Nov _3_	Regression Testing Simulation & Testware	Chapter 16	
10	17	Mon Nov _8_	Testing Usability Performance Testing Scalability Problems State Transition Testing	TBD	Project 6 due Fri at noon (submit at Russ Office).
	18	Wed Nov_10_	Testing Metrics Trends How Test Results Shape Testing		